

What you want do with regular expression?

Regular expressions can be used to check if a string contains the specified search pattern. Regular expressions are used to remove the unnecessary words. It also helps in finding out the correctness of the data and even operations such as finding, replacing and formatting the data is possible using Regular Expressions.

Group/Ranges

.	any single character
(a b)	a or b
[abc]	Range (a or b or c)
[^abc]	Not (a or b or c)
[a-z]	Lower case letter from a to z
[A-Z]	Upper case letter from A to Z
[0-10]	Digit from 0 to 10
\x	Group/subpattern number "x"

Escape Sequences

\	Escape following character
\Q	Begin literal sequence
\E	End literal sequence

Assertions

?=	Lookahead assertion
?!	Negative lookahead
?<=	Lookbehind assertion
?!= or ?<!	Negative lookbehind
?>	Once-only Subexpression
?()	Condition [if then]
?()	Condition [if then else]
?#	Comment

USE

```
import re

string = ""I am Sourabh and my mobile no is 9479864026""

regex = 'd+'

match = re.findall(regex, string)

print(match)
```

Anchors

^	Start of string, or start of line in multi-line pattern
\A	Start of string
\$	End of string, or end of line in multi-line pattern
\Z	End of string
\b	Word boundary
\B	Not word boundary
\<	Start of word
\>	End of word

Characters

\c	Control character
\s	White space
\S	Not white space
\d	Digit
\D	Not digit
\w	Word
\W	Not word
\x	Hexadecimal digit
\O	Octal digit

Pattern Modifiers

g	Global match
i *	Case-insensitive
m	Multiple lines
*	
s *	Treat string as single line
x *	Allow comments and whitespace in pattern
e *	Evaluate replacement
U	Ungreedy pattern
*	

USE

```
import re

string = 'hello 12 hi 89. Its me 343'

pattern = 'd+'

result = re.findall(pattern, string)

print(result)

output#1289343
```

USE

```
import re

phone = "MY phone is +91-94 - 79- 86- 40- 26r "

# Remove anything other than digits

num = re.sub (r'\D', " ", phone)

print ("Phone Num : ", num)
```

USE (cont)

> output Phone num : 9479864026

Quantifiers

*	0 or more	{5}	exactly 5
+	1 or more	{5,}	5 or more
?	0 or 1	{1,5}	1,2,3,4 or 5

Special Characters

\n	New line
\r	Carriage return
\t	Tab
\v	Vertical tab
\f	Form feed
\xxx	Octal character xxx
\xhh	Hex character hh

String Manipulation

\$`	Before matched string
\$'	After matched string
+	Last matched string
\$&	Entire matched string

```
import re
pattern = '^a...s$'
test_string = 'abyss'
result = re.match(pattern, test_string)
if result:
    print("Search successful.")
else:
    print("Search unsuccessful.")
```

String Manipulation

\$`	Before matched string
\$'	After matched string
+	Last matched string
\$&	Entire matched string

posix

[:upper:]	Upper case letters	[A-Z]
[:lower:]	lower case letter	[a-z]
[:alpha:]	alphabets	[a-zA-Z]
[:alnum:]	alphanumeric	[a-zA-z0-9]
[:digit:]	Digits	[0-9]
[:xdigit:]	Hexadecimal	[A-Fa-f0-9]
[:punct:]	Punctuation	[!"#\$%&'()*+,-./:;-<=>?@[\\\]^_`{ }~]
[:blank:]	Space and tab	
[:space:]	Blank characters	
[:cntrl:]	Control characters	
[:graph:]	Printed characters	
[:print:]	Printed characters and spaces	
[:word:]	Digits, letters and underscore	

By sahusourabh



cheatography.com/sahusourabh/

Published 19th July, 2022.
Last updated 19th July, 2022.
Page 2 of 2.

Sponsored by CrosswordCheats.com
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>